Real-time probabilistic fusion of sparse 3D LIDAR and dense stereo

Will Maddern and Paul Newman

Abstract— **Real-time 3D perception is critical for localisation, mapping, path planning and obstacle avoidance for mobile robots and autonomous vehicles. For outdoor operation in real-world environments, 3D perception is often provided by sparse 3D LIDAR scanners, which provide accurate but low-density depth maps, and dense stereo approaches, which require significant computational resources for accurate results. Here, taking advantage of the complementary error characteristics of LIDAR range sensing and dense stereo, we present a probabilistic method for fusing sparse 3D LIDAR data with stereo images to provide accurate dense depth maps and uncertainty estimates in real-time. We evaluate the method on data collected from a small urban autonomous vehicle and the KITTI dataset, providing accuracy results competitive with state-of-the-art stereo approaches and credible uncertainty estimates that do not misrepresent the true errors, and demonstrate real-time operation on a range of low-power GPU systems.**

I. Introduction

Real-time 3D perception is an essential capability of mobile robots and autonomous vehicles, enabling robust localisation, mapping, obstacle avoidance and path planning in challenging real-world environments. The advent of affordable RGBD sensors [1] and 3D LIDAR scanners [2] has led to significant advances over the last decade in the fields of indoor [3], [4] and outdoor [5], [6] robotics respectively. Future robots and autonomous vehicles will continue to benefit from even denser, higher-quality 3D perception and scene understanding.

For outdoor robotics and autonomous vehicles, 3D LIDAR scanners are the most practical sensor for 3D perception, since RGBD sensors are typically short ranged and do not function well in the presence of sunlight [7]. However, a recent trend in low-cost 3D LIDAR scanners is a reduction in point density, with the latest generation of sensors offering fewer than 300K points per second [8]. Another popular alternative is 3D perception from passive stereo cameras, which provide the imagery necessary to compute fully dense depth images for outdoor scenes. However, for accurate results on high-resolution images, the computation can be prohibitively expensive, with many of the top-ranked methods on the KITTI stereo benchmark [9] requiring multiple seconds or minutes of computation time per video frame [10], [11].

In this paper we extend [12] with an efficient probabilistic method of tightly coupling sparse 3D LIDAR data with passive stereo camera images to produce accurate dense depth images in real-time, yielding over 4.5M points per second on a low-power laptop GPU. We seek the best of both worlds and take advantage of the complementary nature of time-of-flight ranging and dense stereo reconstruction to improve 3D perception as follows:

Authors are from the Mobile Robotics Group, Dept. Engineering Science, University of Oxford, UK. {wm,pnewman}@robots.ox.ac.uk

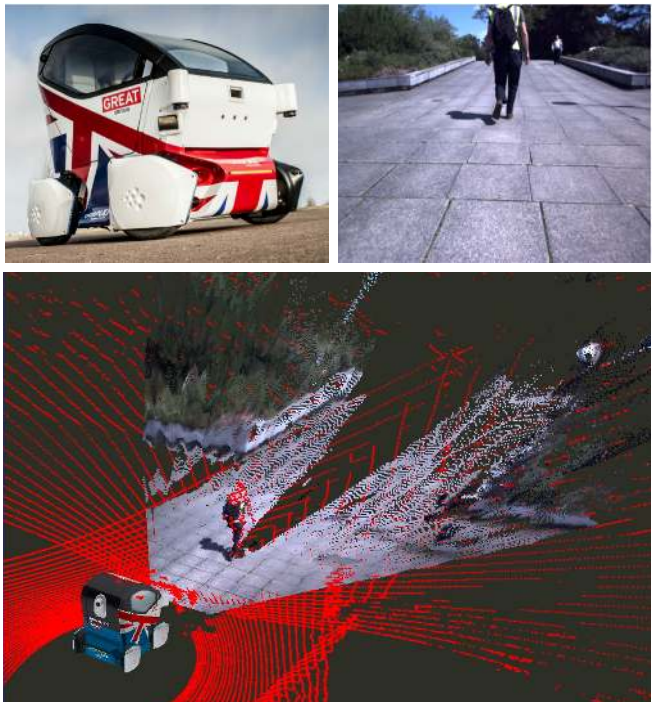

Fig. 1. Real-time fusion of stereo and LIDAR range data for a small autonomous vehicle "Pod" operating in urban environments. The vehicle (top left) is equipped with multiple low-cost 3D LIDAR scanners and a stereo camera. Images from the camera (top right) are fused in real-time with sparse 3D LIDAR points (bottom, red) to form a dense depth map (bottom, coloured) suitable for online localisation, mapping, path planning and obstacle avoidance.

Reduced disparity search: a sparse 3D LIDAR prior significantly reduces the range of valid disparities to search for in the stereo image, reducing computation time.

Complementary error characteristics: dense stereo-only depth errors increase quadratically with increased range [13], but sparse 3D LIDAR point errors increase linearly with distance.

For autonomous vehicle applications it is also important that the uncertainty of the 3D perception system is accounted for in the planning stage, in order to make conservative decisions to increase safety. Importantly, our probabilistic approach provides real-time sub-pixel estimates of the 3D position and uncertainty of all points.

We evaluate the method using both data collected in urban environments from an outdoor autonomous vehicle platform

(see Fig. 1) and the KITTI stereo benchmark dataset [14]. We demonstrate real-time operation on a range of consumer GPU hardware with different power requirements, towards the goal of low-cost dense 3D perception for autonomous vehicles.

II. Related Work

LIDAR-stereo fusion is commonplace in a number of related fields; in particular the combination of airborne LIDAR mapping and satellite imagery for terrain reconstruction [15], [16], [17]. The goal is to maximise the accuracy of the reconstruction for geographic and scientific purposes, typically achieved by large-scale offline optimisation approaches which are not suitable for real-time operation.

The fusion of LIDAR and stereo imagery for robotics applications has been demonstrated in 2D [18] and 3D [19], [20], with both taking advantage of the complementary nature of the sensors. Notably, [20] combines Velodyne HDL64-E LIDAR data with stereo imagery in a dynamic programming framework [21] and illustrates the advantages gained in terms of reduced computation time, but only provides qualitative accuracy results. A similar approach is presented in [22], where data from a time-of-flight imaging sensor is fused with a stereo camera.

Real-time dense stereo approaches for automotive applications have also progressed in recent years, with methods such as [23], [12] providing results with sub-10% error rates on the KITTI benchmark [9]. The accuracy of Semi-Global Matching (SGM) in particular compares favourably with LIDAR for digital surface model (DSM) generation [24]. However, methods that further increase the accuracy of stereo-only reconstruction rely on additional prior knowledge [10], [14] or deep learning [11], neither of which permit real-time operation on current computing hardware. Additionally, none of the above approaches directly estimate 3D uncertainty, nor is the estimator credibility [25] assessed by the KITTI benchmark.

III. LIDAR-Stereo Fusion

In the following section we outline our approach to real-time probabilistic LIDAR-stereo fusion for 3D perception. Following the approach in [12] we factor the disparity estimation problem into two independent distributions:

$$p\left(d_i \mid \mathbf{o}_i^{(l)}, \mathbf{o}_i^{(r)}, \mathbf{S}\right) \propto p\left(\mathbf{o}_i^{(r)} \mid d_i, \mathbf{o}_i^{(l)}\right) p\left(d_i \mid \mathbf{o}_i^{(l)}, \mathbf{S}\right) \tag{1}$$

where d_i is the disparity at pixel i between left and right images, $\mathbf{o}_i^{(l,r)}$ is an observation of a feature at pixel i encoding both the feature position and descriptor in the left or right image, and $\mathbf{S}$ is a set of sparse support points produced by either LIDAR ranges, sparse stereo feature matching or a combination of the two. By factoring the distribution the estimation of $p\left(d_i \mid \mathbf{o}_i^{(l)}, \mathbf{o}_i^{(r)}, \mathbf{S}\right)$ is decoupled into two independent stages:

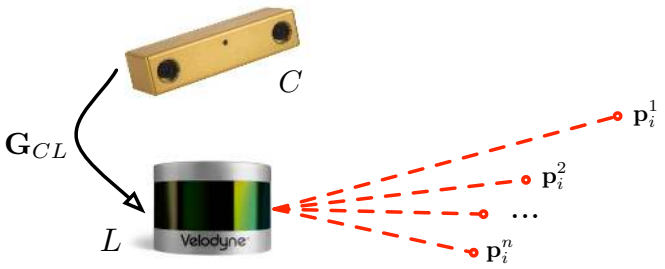

Fig. 2. Extrinsics of LIDAR and stereo system. The matrix $\mathbf{G}_{CL}$ transforms points $\mathbf{p}_i^j$ from the LIDAR frame L into the stereo camera frame C. A scanning 3D LIDAR will sample n points at different elevations for each azimuth index i.

Disparity prior estimation: using sparse 3D LIDAR points and/or sparse robustly matched keypoints, interpolate disparities with a triangle mesh over the whole image.

Disparity refinement: for each pixel, sample appearance descriptors from a small range of disparities and fit a Gaussian to the resulting posterior distribution.

The following sections describe each of these steps in detail along with a pyramid interpolation approach to produce a fully dense probabilistic disparity image.

A. Disparity Prior

We first seek to represent the prior distribution $p\left(d_i \mid \mathbf{o}_i^{(l)}, \mathbf{S}\right)$ in terms of the support points $\mathbf{S}$ - which could be from LIDAR, sparse stereo matching or both. By defining a mean disparity function $\hat{\mu}_*\left(\mathbf{o}_i^{(l)}, \mathbf{S}\right)$ and standard deviation function $\hat{\sigma}_*\left(\mathbf{o}_i^{(l)}, \mathbf{S}\right)$ generated by interpolating information from support points $\mathbf{S}$ at pixel i, we can express the prior distribution as a Gaussian:

$$p\left(d_i \mid \mathbf{o}_i^{(l)}, \mathbf{S}\right) \propto \exp\left[-\frac{\left(d_i - \hat{\mu}_*\left(\mathbf{o}_i^{(l)}, \mathbf{S}\right)\right)^2}{\hat{\sigma}_*\left(\mathbf{o}_i^{(l)}, \mathbf{S}\right)^2}\right] \tag{2}$$

We follow the approach in [12] to compute sparse robust support points from feature matches between left and right stereo images, but substitute a fast triangulation on a regular grid for the slower but more exact Delaunay triangulation between supports. The stereo prior mean function $\hat{\mu}_C(\cdot)$ then becomes a linear interpolation between support points at the vertices of the triangle. Since the stereo camera disparity prior is obtained from support points estimated in disparity space, the stereo prior standard deviation function $\hat{\sigma}_C(\cdot)$ is constant across the entire image:

$$\hat{\sigma}_C\left(\mathbf{o}_i^{(l)}, \mathbf{S}\right) = \sigma_C \tag{3}$$

where σ_C is the support point matching standard deviation in disparity space.

To generate support points from the LIDAR sensor, each LIDAR range measurement $\mathbf{p}_i^j$ must be transformed into the camera frame C as follows:

$$ {}^{C}\mathbf{p}_i^j = \mathbf{K}\mathbf{G}_{CL}\mathbf{p}_i^j \tag{4} $$

where $\mathbf{K}$ is the camera intrinsic matrix and $\mathbf{G}_{CL}$ is the extrinsic calibration between the camera C and the LIDAR L as illustrated in Fig. 2. To triangulate individual LIDAR points into a mesh, we take advantage of the rotational scanning nature of 3D LIDAR scanners. For a scanner with n individual laser beams indexed in elevation by j, we assemble candidate camera-frame triangles ${}^{C}\mathbf{t}_{a,b}$ from two sequential azimuth indices $i, i+1$:

$$ \begin{aligned} {}^{C}\mathbf{t}_a &= \left\{ {}^{C}\mathbf{p}_i^j, {}^{C}\mathbf{p}_{i+1}^j, {}^{C}\mathbf{p}_{i+1}^{j+1} \right\} \\ {}^{C}\mathbf{t}_b &= \left\{ {}^{C}\mathbf{p}_i^j, {}^{C}\mathbf{p}_i^{j+1}, {}^{C}\mathbf{p}_{i+1}^{j+1} \right\} \end{aligned} \tag{5} $$

If all triangles are preserved, the prior may connect points on different objects, leading to undesirable interpolation over discontinuities in the scene. We adopt a simple approach to prune excessively large triangles by removing those with edges longer than a threshold δ; for urban environments we found $\delta = 1.0$m to be an acceptable threshold for meshing without interpolating triangles between different objects. In the case where azimuth and elevation indices i, j are not available, triangles can be formed using a Delaunay triangulation over the camera-frame points at a higher computational cost.

To take advantage of the linear distance error characteristics of LIDAR ranging, the LIDAR prior standard deviation function $\hat{\sigma}_L(\cdot)$ incorporates the prior disparity mean function $\hat{\mu}_L(\cdot)$ as follows:

$$ \hat{\sigma}_L\left(\mathbf{o}_i^{(l)}, \mathbf{S}\right) = \frac{\hat{\mu}_L\left(\mathbf{o}_i^{(l)}, \mathbf{S}\right)}{f_x b}\sigma_L \tag{6} $$

where f_x is the camera focal length, b is the baseline between the left and right camera and σ_L is the LIDAR range error standard deviation, obtained from the sensor specifications. This results in reduced disparity prior uncertainty at longer ranges, where the LIDAR sensor accuracy is typically superior to the stereo camera.

Interpolating the prior mean $\hat{\mu}_*(\cdot)$ and standard deviation $\hat{\sigma}_*(\cdot)$ functions across the image can be performed using standard rasterising methods for rendering a triangle mesh. If multiple priors exist at pixel i (in the case of both a stereo and LIDAR prior), the prior with the lowest standard deviation $\hat{\sigma}_*(\cdot)$ is selected. Fig. 3 illustrates a stereo, LIDAR and combined prior for a sample image.

B. Disparity Refinement

After the prior mean $\hat{\mu}_*(\cdot)$ and standard deviation $\hat{\sigma}_*(\cdot)$ are computed for each pixel i, the disparity estimates d_i can be refined using fine-grained *appearance* information from the stereo camera images. We approximate the likelihood term from Eq. 1 by a Laplace distribution as in [12] as follows:

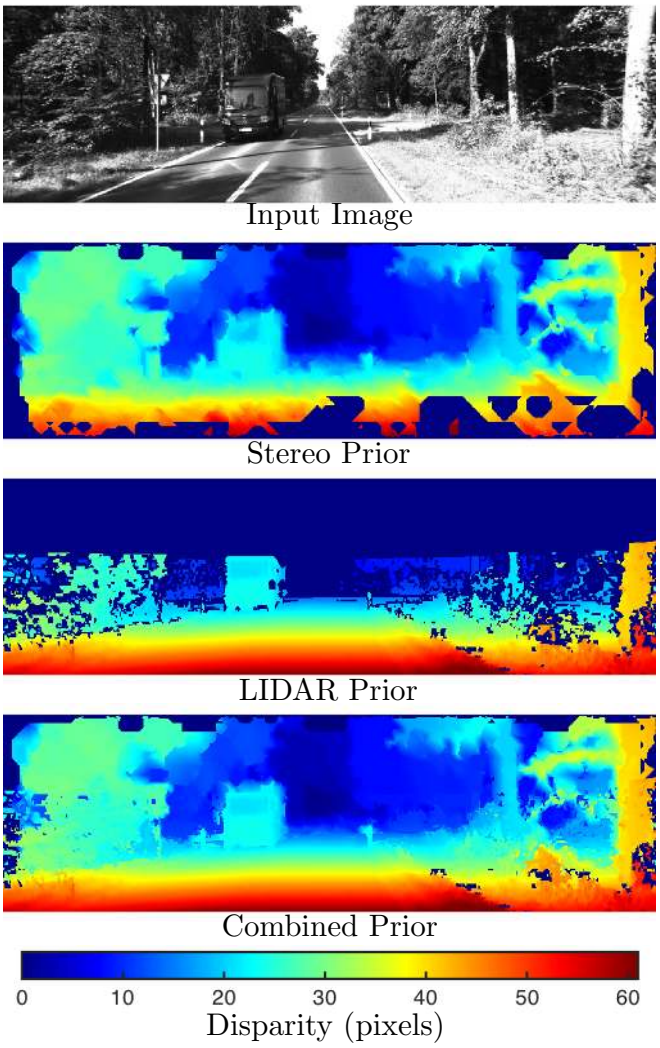

Fig. 3. Disparity prior estimated using stereo and LIDAR sensors. The "Stereo" method uses only sparse points matched from stereo images as support points $\mathbf{S}$; the "LIDAR" method uses only 3D LIDAR points as the support points $\mathbf{S}$; the "Combined" method uses both the above priors. The Stereo prior provides good coverage except in locations with low texture (e.g. road surfaces); the LIDAR prior is less noisy than the stereo but is sparser and covers less of the overall field of view; the combined prior yields the best of both sensors.

$$ p\left(\mathbf{o}_i^{(r)} \mid d_i, \mathbf{o}_i^{(l)}\right) \propto \exp\left[-\beta\left\|\mathbf{f}_i^{(l)} - \mathbf{f}_i^{(r)}(d_i)\right\|_1\right] \tag{7} $$

where $\mathbf{f}_i^{(l)}$ is an appearance descriptor extracted at location i in the left image, $\mathbf{f}_i^{(r)}(d_i)$ is the corresponding descriptor in the right image displaced by d_i pixels along the epipolar line, and β is a weighting factor for the descriptor cost. For the appearance descriptor we use a simplified 16-element version of the gradient-based descriptor presented in [12].

Instead of performing maximum a-posteriori estimation to obtain only the optimal integer disparity value as in [12], we seek to capture the posterior of Eq. 1 with a sampled Gaussian distribution. The weighted mean disparity $\bar{d}_i$ is computed as follows:

$$\bar{d}_i = \frac{\sum_k d_k \cdot p\left(d_k \mid \mathbf{o}_k^{(l)}, \mathbf{o}_k^{(r)}, \mathbf{S}\right)}{\sum_k p\left(d_k \mid \mathbf{o}_k^{(l)}, \mathbf{o}_k^{(r)}, \mathbf{S}\right)} \tag{8}$$

where $p\left(d_k \mid \mathbf{o}_k^{(l)}, \mathbf{o}_k^{(r)}, \mathbf{S}\right)$ is the product of the terms in Eq. 2 and Eq. 7 as per Eq. 1. To fully exploit the prior the sampled disparities d_k for each pixel i are limited to those that satisfy the following inequality:

$$\left\| d_k - \hat{\mu}_* \left(\mathbf{o}_k^{(l)}, \mathbf{S}\right) \right\| \leq 3\hat{\sigma}_* \left(\mathbf{o}_k^{(l)}, \mathbf{S}\right) \tag{9}$$

hence only disparities within 3 standard deviations of the prior mean $\hat{\mu}_*(\cdot)$ are sampled. For the same reasons noted in [12] this sampling approach is independent for each pixel i and can be performed in parallel. The disparity variance $\bar{\sigma}_i^2$ can be estimated for each pixel using the same k samples as follows:

$$\bar{\sigma}_i^2 = \frac{\sum_k d_k^2 \cdot p\left(d_k \mid \mathbf{o}_k^{(l)}, \mathbf{o}_k^{(r)}, \mathbf{S}\right)}{\sum_k p\left(d_k \mid \mathbf{o}_k^{(l)}, \mathbf{o}_k^{(r)}, \mathbf{S}\right)} - \left[\frac{\bar{d}_i}{\sum_k p\left(d_k \mid \mathbf{o}_k^{(l)}, \mathbf{o}_k^{(r)}, \mathbf{S}\right)}\right]^2 \tag{10}$$

In scenarios when parts of the scene are occluded from one of the two cameras, or where the prior has interpolated between support points that are not connected in the scene, estimating disparities from the left image alone will yield incorrect results. Hence we perform the estimation of Eq. 7 twice: once for left-to-right matches and once for right-to-left matches. After obtaining both disparity estimates $\bar{d}_i^{(l)}, \bar{d}_i^{(r)}$ and error estimates $\bar{\sigma}_i^{(l)}, \bar{\sigma}_i^{(r)}$ we can assess the consistency by computing the Mahalanobis distance between the estimates; if the distance exceeds a threshold ϕ, the disparity estimate $\bar{d}_i$ is deemed to be an outlier and the pixel i is marked as invalid. In practice this results in images with approximately 60-70% density (counting only valid disparities); the following section details the pyramid-based interpolation process to reach 100% density.

C. Pyramid Interpolation

To manage invalid disparities we adopt a two-stage downscale-upscale pyramid approach with P levels. We implement a downscale operator that accepts a 2×2 block of possibly invalid disparity and uncertainty measurements, and compute a single combined disparity and uncertainty. The combined disparity $\bar{d}_c$ for each block is computed by averaging the mean of each of the N valid disparities weighted by their inverse variance as follows:

$$\bar{d}_c = \frac{\sum_k^N \bar{d}_k \cdot \bar{\sigma}_k^{-2}}{\sum_k^N \bar{\sigma}_k^{-2}} \tag{11}$$

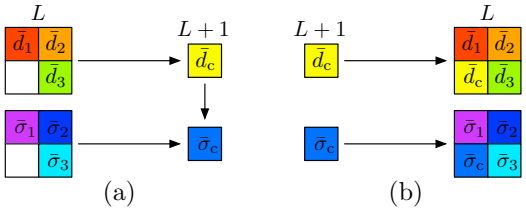

Fig. 4. Pyramid interpolation (a) downscale and (b) upscale operators. At pyramid level L, there may be invalid or missing disparities in each 2×2 block. The combined disparity $\bar{d}_c$ and uncertainty $\bar{\sigma}_c$ are computed from the valid disparities and uncertainties using Eq. 11 and 12 to form pyramid level $L+1$, ensuring strictly increasing variance with higher pyramid levels; this is repeated for each level. The combined values are then successively substituted back into each level L to fill in missing values with increased uncertainty.

The combined disparity for any block consisting only of invalid disparities (i.e. $N = 0$) is marked as invalid. The combined variance $\bar{\sigma}_c^2$ for each block is computed as the average second non-central moment about the combined disparity $\bar{d}_c$ as follows:

$$\bar{\sigma}_c^2 = \frac{1}{N} \sum_k^N \left[\left(\bar{d}_k - \bar{d}_c\right)^2 + \bar{\sigma}_k^2 \right] \tag{12}$$

This approach ensures that the variance strictly increases with successive pyramid levels, and significantly in areas where disparities are interpolated across object boundaries. The upscale process is performed by simply substituting any invalid disparity value with the corresponding valid combined disparity from the next pyramid level, and repeating for each pyramid level P. This ensures that all valid disparities estimated in Section III-B are preserved, and invalid values are replaced with interpolated disparities with correspondingly higher uncertainties, resulting in a fully dense disparity image. The downscale and upscale operators are illustrated in Fig. 4, and Fig. 5 illustrates the full pyramid downscale and upscale approach on sample images.

IV. GPU Implementation

The method presented in Section III has been structured for implementation on a modern GPU with minimal modification. For automotive or mobile applications, GPUs offer far higher computational throughput for the same power consumption as an equivalent CPU [26]. For portability the method is implemented in GL Shader Language[1] for a modern OpenGL implementation.

The disparity prior interpolation in Section III-A is performed using the hardware rasterisation pipeline in OpenGL, where input triangles connecting support points $\mathbf{S}$ are sampled on a regular grid the size of the image to produce a dense disparity prior. The resulting prior is stored in texture memory on the GPU.

The disparity refinement in Section III-B is implemented as a shader that samples the disparity prior texture to compute the bound $3\hat{\sigma}_* \left(\mathbf{o}_k^{(l)}, \mathbf{S}\right)$, then iteratively samples descriptors $\mathbf{o}_k^{(l,r)}$ along the epipolar line of the left and right images

[1] https://www.opengl.org/documentation/glsl/

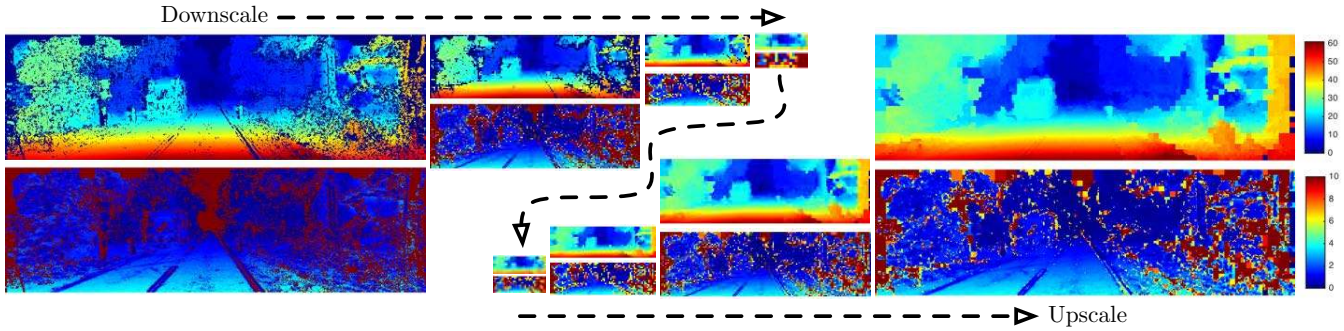

Fig. 5. Pyramid interpolation downscale and upscale for a sample image from the KITTI dataset. The disparity and uncertainty estimates are successively downscaled (left) using the operator in Fig. 4(a) until reaching the maximum number of pyramid levels. The estimates are then upscaled using the operator in Fig. 4(b) until reaching the original resolution. Note the significantly increased density of the disparity image after the pyramid interpolation, and increased variance in locations where disparities are interpolated. The process only takes 1-2ms on a low-power laptop GPU.

within the bound. From these samples the shader computes the estimated disparity $\bar{d}_i$ and variance $\bar{\sigma}_i^2$ for a single pixel, and hence the GPU can simultaneously evaluate as many disparity estimates as it has processor cores.

The pyramid interpolation method in Section III-C is implemented using a set of texture buffers of decreasing resolution. Each successive downscale operation samples N valid disparities from the current resolution to compute the corresponding combined disparity in the next pyramid level. Each successive upscale operation populates only the invalid disparities with those from the next pyramid level. This approach is highly efficient and typically only adds 1-2ms to the entire process, even for high resolution images.

As will be illustrated in Section VI, the GPU implementation enables dense depth map computation from both a LIDAR and stereo prior in under 100ms on a low-power laptop GPU, equal to the 10Hz frame rate of the KITTI stereo benchmark images.

V. Experimental Setup

We evaluate the system on two different platforms in different environments: the small autonomous vehicle "Pod" pictured in Fig. 1 and the KITTI dataset [9]. The Pod is equipped with two Velodyne HDL-32E 32-beam LIDAR scanners and a Bumblebee XB3 stereo camera, capturing 640×480 resolution images at 16Hz. The KITTI dataset provides Velodyne HDL-64E 64-beam LIDAR data and 1232×368 resolution stereo images captured at 10Hz.

For the KITTI evaluation we make use of the raw data development kit [27] to extract Velodyne scans and associate them with stereo images. Since the Velodyne data is not supplied with azimuth and elevation indices i, j, we perform a Delaunay triangulation on points projected into the camera frame, which adds approximately 10ms additional processing time. We also do not compensate for the motion of the vehicle during the LIDAR scan, instead relying on the disparity refinement process to correct for motion in the scene.

It is challenging to generate ground truth for outdoor disparity estimation, especially when LIDAR scans often form the basis of ground truth for depth maps. For the KITTI dataset evaluation we use the provided Stereo12 ground truth [9], formed by accumulating multiple motion-compensated Velodyne scans along with manually annotated CAD models of moving vehicles. This results in a far denser disparity ground truth than a single raw scan. Of the 200 images supplied with the KITTI stereo evaluation benchmark, 141 have associated raw Velodyne scans. For the Pod dataset no ground truth is available, so we present only qualitative results on tens of kilometres driven in a mixed pedestrian and vehicle urban environment.

TABLE I
Algorithm Parameters

Symbol	Parameter	Value
σ_C	Stereo Prior Uncertainty	3 pix
σ_L	LIDAR Prior Uncertainty	0.1m
δ	LIDAR Triangle Max Edge Length	1.0m
β	Descriptor Cost Factor	0.25
ϕ	Left-Right Verification Threshold	2.0
P	Pyramid Interpolation Levels	6

We test computation time on two different computing platforms representing a range of power consumption and portability requirements. The first is a 2015 Macbook Pro equipped with a Core i7 processor and an AMD Radeon R9 M370X GPU, with 640 processors at 800MHz drawing 50W. The second is a high-end desktop equipped with a Core i7 processor and an AMD Radeon R9 295x2 GPU, with 5632 processors at 1018MHz drawing 450W. Both systems are capable of producing dense depth maps from LIDAR and stereo data at frame rate on both platforms.

Table I lists the algorithm parameters used for evaluation.

VI. Results

In this section we evaluate the disparity accuracy, estimator credibility and computational requirements of the proposed LIDAR-stereo fusion approach.

A. Disparity Estimation

An analysis of the disparity estimation accuracy on the KITTI benchmark is presented in Table II; examples of the six different methods evaluated are illustrated in Fig. 6. Errors were calculated using the software provided with the

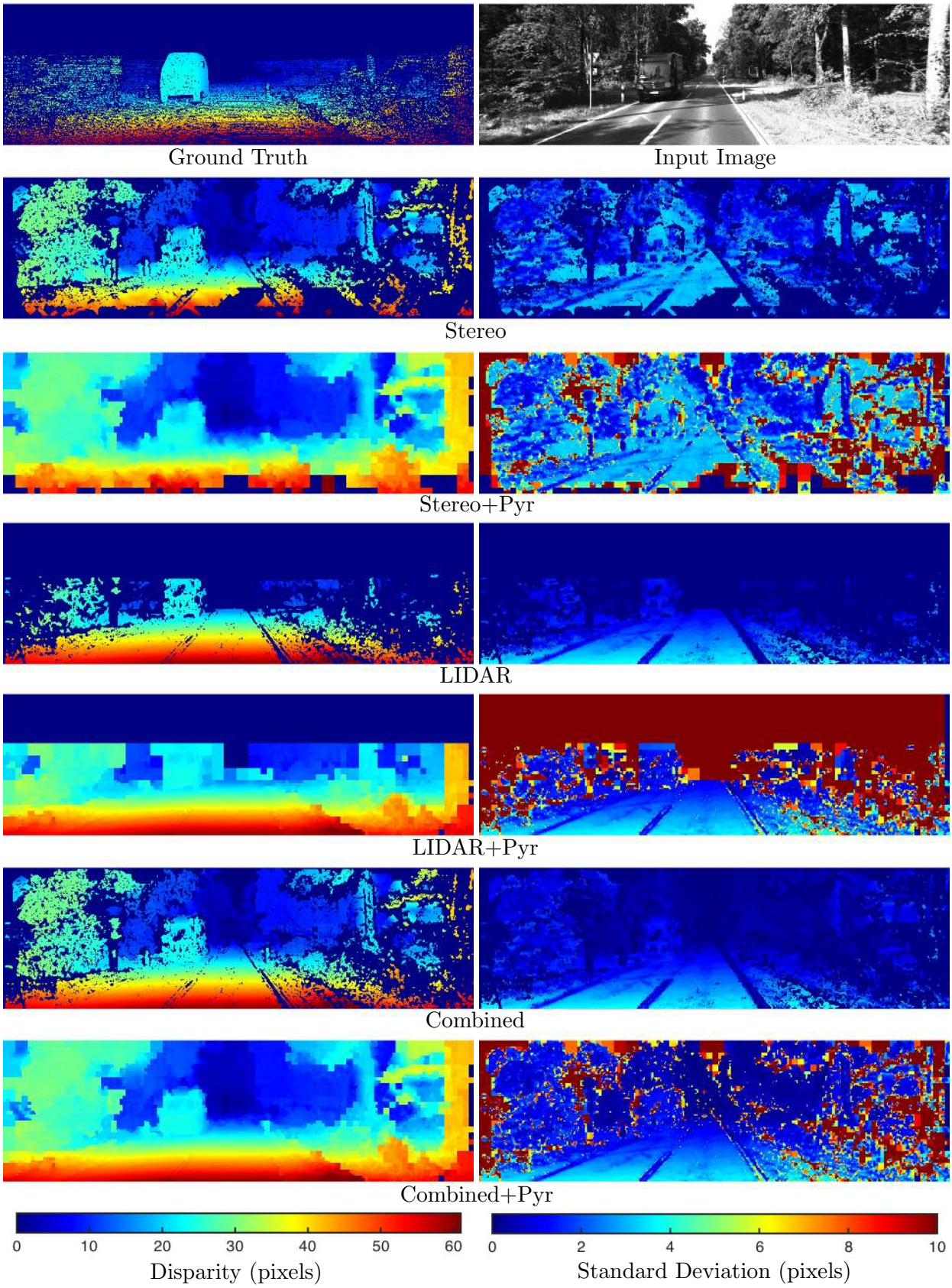

Fig. 6. Disparities and uncertainties estimated using the six methods in Table II for a sample image in the KITTI dataset. The "Stereo", "LIDAR" and "Combined" methods refer to the priors in Fig. 3. All approaches with "Pyr" use the pyramid interpolation method of Section III-C to fill gaps in the disparity and uncertainty images. The Stereo method provides a valid prior for most of the image but results in higher uncertainties. The LIDAR method has reduced uncertainties at longer ranges, but only provides a prior for the bottom 2/3 of the image. The Combined method yields the best overall performance.

TABLE II

KITTI DATASET PERFORMANCE

Method	Background		Foreground		All		Density
	Err	ANEES	Err	ANEES	Err	ANEES	
Stereo	9.39%	1.07	8.01%	1.86	9.19%	1.19	67.20%
Stereo+Pyr	17.78%	1.34	16.12%	2.37	17.51%	1.51	94.31%
LIDAR	0.35%	0.29	5.24%	2.11	0.94%	0.51	68.52%
LIDAR+Pyr	7.17%	0.37	15.65%	2.02	8.51%	0.63	99.24%
Combined	1.37%	0.46	6.27%	2.55	1.99%	0.73	79.47%
Combined+Pyr	4.55%	0.68	13.18%	2.75	5.91%	1.01	99.62%

KITTI Stereo Development Kit[2]. While the accuracy of the stereo-only approach is comparable with the results of [12], the addition of LIDAR data significantly reduces the error rates to less than 6% for the dense Combined+Pyr approach. Both the LIDAR and Combined methods provide lower accuracy on foreground objects than background; this is due to a combination of poor raw LIDAR returns on vehicle surfaces and the relative motion of the vehicles during the scan. Performing pyramid interpolation significantly increases the density of the resulting disparity images, from 60-70% to more than 99% in the LIDAR and Combined methods, at the cost of slightly increased errors in the interpolated areas. At the time of writing the Combined+Pyr results are competitive with the top 10 results on the KITTI 2015 stereo benchmark while retaining real-time performance.

Qualitative results on the Pod datasets are illustrated in Fig. 7. Empirically, the method produces accurate reconstructions of different surfaces, including roads, pavements, vegetation and buildings, as well as dynamic objects such as pedestrians and vehicles.

B. Estimator Credibility

For mobile robot and autonomous vehicle applications it is important that the disparity uncertainty estimates $\bar{\sigma}_i$ do not underestimate the true errors, as this may lead to overconfident behaviour in uncertain environments. To evaluate the disparity uncertainty estimates, we compute the average normalised estimation error squared (ANEES), which characterises the consistency of a state estimator [25]. The ANEES score $\bar{\epsilon}$ is computed as follows:

$$\bar{\epsilon} = \frac{1}{M} \sum_{k}^{M} \left[\frac{\bar{d}_k - d_k}{\bar{\sigma}_k} \right]^2 \tag{13}$$

where $\bar{d}_k$ and $\bar{\sigma}_k$ are the estimated disparity and standard deviation respectively, and d_k is the true disparity from ground truth. Over all k the set of samples will follow a chi-squared distribution, which when divided by the number of samples M will yield the ANEES score $\bar{\epsilon}$, which for a credible estimator will equal the dimension of the state vector (i.e. 1). If the ANEES score $\bar{\epsilon} < 1$ for the disparity estimation problem, then the estimator is *conservative*; i.e. it overestimates the uncertainty of the error distribution. If $\bar{\epsilon} > 1$ then the estimator is *overconfident*; the true error is greater than that estimated by the system.

The ANEES scores for each disparity estimation method are shown in Table II. The Stereo and Stereo+Pyr methods are overconfident for all pixels; methods using the LIDAR sensor are typically conservative for background pixels but overconfident on foreground pixels. The Combined+Pyr approach on all pixels yields the overall most credible estimate of the disparity uncertainty, with an ANEES score of 1.01.

[2] http://www.cvlibs.net/datasets/kitti/eval_scene_flow.php?benchmark=stereo

C. Computation Time

TABLE III

KITTI DATASET COMPUTATION TIME

Method	AMD R9 M370X 50W	AMD R9 295x2 450W
Stereo	84.91ms	19.81ms
Stereo+Pyr	86.975ms	21.39ms
Lidar	33.63ms	8.69ms
Lidar+Pyr	34.97ms	9.83ms
Combined	96.44ms	23.01ms
Combined+Pyr	98.05ms	24.35ms

Mean processing times on the two GPUs described in Section V for KITTI dataset images are presented in Table III. Both GPUs produced depth maps in under 100ms, providing real-time performance on the KITTI dataset where the images and LIDAR data are collected at 10Hz. Using only LIDAR information for the disparity prior significantly reduces the computation time, providing depth maps at 30Hz on the mobile GPU and 100Hz on the desktop GPU. Although the desktop GPU provides approximately 10 times the computational power of the laptop GPU, due to memory copy and draw overheads it only provides approximately 4 times the depth map rate for the most accurate Combined+Pyr method. However, this could be useful in scenarios where a vehicle is fitted with multiple stereo cameras (e.g. 4 cameras providing a panoramic view combined with a single Velodyne LIDAR scanner, providing full 360° coverage dense depth maps at 10Hz).

VII. CONCLUSIONS

In this paper we presented a probabilistic approach to LIDAR-stereo fusion in real-time. By propagating uncertainty estimates through a disparity prior and refinement stage and structuring the method for optimal implementation on a GPU, we are able to provide accurate sub-pixel disparity and uncertainty estimates at camera frame rate using a laptop GPU. The accuracy of our method is competitive with state-of-the-art approaches on the KITTI dataset, and the method provides credible uncertainty estimates that do not misrepresent the true error. We hope these results pave the way towards low-cost dense 3D perception for future mobile robots and autonomous vehicles operating in unconstrained, real-world environments.

ACKNOWLEDGMENTS

Will Maddern is supported by an EPSRC Programme Grant EP/M019918/1. Paul Newman is supported by an EPSRC Leadership Fellowship EP/I005021/1.

REFERENCES

[1] Microsoft Corporation. Microsoft Kinect Product Description. [Online]. Available: http://www.xbox.com/kinect/

[2] Velodyne Lidar. Velodyne HDL-64E Product Description. [Online]. Available: http://velodynelidar.com/hdl-64e.html

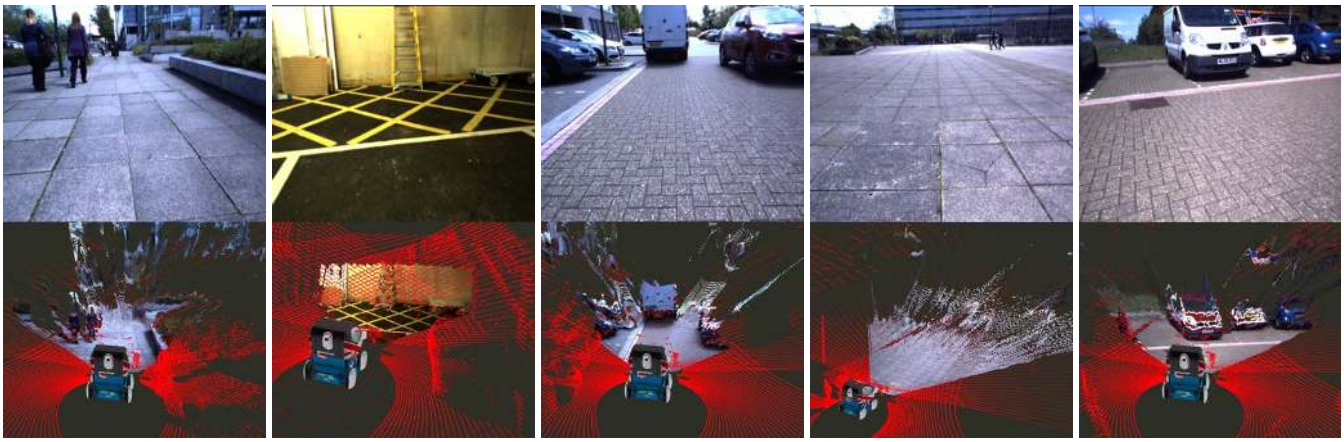

Fig. 7. Example real-time 3D perception for the "Pod" autonomous vehicle operating in an urban environment. The Bumblebee XB3 stereo images (top) are fused with sparse 3D LIDAR data from the two Velodyne HDL32-E sensors mounted on the roof (bottom, red points). The method provides accurate dense 3D reconstructions (bottom, coloured points) on a variety of objects, including road and pavement surfaces, vehicles and pedestrians.

[3] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, "RGB-D mapping: Using depth cameras for dense 3d modeling of indoor environments," in *In the 12th International Symposium on Experimental Robotics (ISER*. Citeseer, 2010.

[4] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "KinectFusion: Real-time dense surface mapping and tracking," in *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*. IEEE, 2011, pp. 127–136.

[5] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer, *et al.*, "Autonomous driving in urban environments: Boss and the urban challenge," *Journal of Field Robotics*, vol. 25, no. 8, pp. 425–466, 2008.

[6] R. W. Wolcott and R. M. Eustice, "Fast LIDAR localization using multiresolution gaussian mixture maps," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 2814–2821.

[7] S. M. Abbas and A. Muhammad, "Outdoor RGB-D SLAM performance in slow mine detection," in *Robotics; Proceedings of ROBOTIK 2012; 7th German Conference on*. VDE, 2012, pp. 1–6.

[8] Velodyne Lidar. Velodyne VLP-16 Product Description. [Online]. Available: http://velodynelidar.com/vlp-16.html

[9] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 3354–3361.

[10] F. Guney and A. Geiger, "Displets: Resolving stereo ambiguities using object knowledge," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4165–4175.

[11] J. Zbontar and Y. LeCun, "Stereo matching by training a convolutional neural network to compare image patches," *arXiv preprint arXiv:1510.05970*, 2015.

[12] A. Geiger, M. Roser, and R. Urtasun, "Efficient large-scale stereo matching," in *Computer Vision–ACCV 2010*. Springer, 2010, pp. 25–38.

[13] D. Gallup, J.-M. Frahm, P. Mordohai, and M. Pollefeys, "Variable baseline/resolution stereo," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.

[14] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[15] T. Schenk and B. Csathó, "Fusion of LIDAR data and aerial imagery for a more complete surface description," *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences*, vol. 34, no. 3/A, pp. 310–317, 2002.

[16] L. C. Chen, T.-A. Teo, Y.-C. Shao, Y.-C. Lai, and J.-Y. Rau, "Fusion of LIDAR data and optical imagery for building modeling," *International Archives of Photogrammetry and Remote Sensing*, vol. 35, no. B4, pp. 732–737, 2004.

[17] G. Sohn and I. Dowman, "Data fusion of high-resolution satellite imagery and LiDAR data for automatic building extraction," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 62, no. 1, pp. 43–63, 2007.

[18] K. Nickels, A. Castano, and C. Cianci, "Fusion of lidar and stereo range for mobile robots," in *Int. Conf. on Advanced Robotics*, 2003.

[19] A. Harrison and P. Newman, "Image and sparse laser fusion for dense scene reconstruction," in *Proceedings of the 7th International Conference on Field and Service Robotics*, Cambridge, Massachusetts, July 2009.

[20] H. Badino, D. Huber, and T. Kanade, "Integrating LIDAR into stereo for fast and improved disparity computation," in *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2011 International Conference on*. IEEE, 2011, pp. 405–412.

[21] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International journal of computer vision*, vol. 47, no. 1-3, pp. 7–42, 2002.

[22] V. Gandhi, J. Čech, and R. Horaud, "High-resolution depth maps based on TOF-stereo fusion," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 4742–4749.

[23] H. Hirschmüller, "Stereo processing by semiglobal matching and mutual information," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 30, no. 2, pp. 328–341, 2008.

[24] S. Gehrke, K. Morin, M. Downey, N. Boehrer, and T. Fuchs, "Semi-global matching: An alternative to LIDAR for DSM generation," in *Proceedings of the 2010 Canadian Geomatics Conference and Symposium of Commission I*, 2010.

[25] X. R. Li, Z. Zhao, and V. P. Jilkov, "Estimator's credibility and its measures," in *Proc. IFAC 15th World Congress*, 2002.

[26] S. Huang, S. Xiao, and W.-c. Feng, "On the energy efficiency of graphics processing units for scientific computing," in *Parallel & Distributed Processing, 2009. IPDPS 2009. IEEE International Symposium on*. IEEE, 2009, pp. 1–8.

[27] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *The International Journal of Robotics Research*, 2013.